

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Previously Presented) A data processing method for application layer based on a predetermined protocol composed of at least a lower layer and an application layer, the method comprising the steps of:

receiving a primitive from an upper application software;
generating a communication cycle identifier (CycleID) according to the primitive;
generating a service description according to the primitive and the communication cycle identifier (CycleID);
composing an application layer protocol data unit (APDU) including the primitive;
and
transmitting the APDU to the lower layer.

2. -11. (Canceled)

12. (Original) The method of claim 1 further comprises the step of:

completing a communication cycle.

13. (Previously Presented) The method of claim 12, wherein the completion step comprises the sub-step of:

making a first decision regarding whether the network layer completing primitive (NLCompleted) from the lower layer is received for a service time out (SvcTimeOut) from the service execution.

14. (Previously Presented) The method of claim 13, wherein if, in the first decision step, the network layer completing primitive (NLCompleted) is received for the service time out (SvcTimeOut), the communication cycle number (CycleNo) included in the service description is

reduced by a predetermined value, and if not, the communication cycle number (CycleNo) is set to '0'.

15. (Previously Presented) The method of claim 13, wherein the completion step further comprises the sub-step of:

making a second decision regarding whether the communication cycle number (CycleNo) included in the service description is '0'.

16. (Previously Presented) The method of claim 15, wherein if, in the second decision step, the communication cycle number (CycleNo) is '0', the service description is deleted, and if not, the communication cycle identifier (CycleID) is generated or the APDU is composed.

17. (Previously Presented) The method of claim 13, wherein the completion step further comprises the step of:

if, in the first decision step of the completion, the network layer completing primitive (NLCompleted) is received before the service time out (SvcTimeOut) lapses or if the service time out (SvcTimeOut) lapses, deleting the service description.

18. (Canceled)

19. (Currently Amended) The method of ~~one of claims~~claim 1, 2, 5, 12 to 16, further comprises the step of:

after the service description is deleted, transmitting an application layer completing primitive (ALCompleted) to the application software.

20. (Currently Amended) The method of ~~one of claims~~claim 1, 2, 5, 12 to 16, further comprises the step of:

after the communication cycle identifier (CycleID) is generated, searching a pre-stored service description corresponding to the generated communication cycle identifier (CycleID), so as to decide, based on the search result, whether to generate the service description.

21. (Currently Amended) The method ~~of one of claims 1, 12 to 16~~, further comprises the steps of:

after the primitive is received, separating a communication cycle based on an application layer service code (ALSvcCode) included in the primitive; and

generating a request or a notification message according to the separated communication cycle,

wherein the communication cycle is separated and the request or the notification message is generated during the composition step of the APDU.

22. - 28. (Canceled)

29. (Currently Amended) The method of ~~one of~~ claims ~~28-1, 12, and 16~~, further comprises the steps of:

after the communication cycle identifier (CycleID) is generated, searching a pre-stored service description corresponding to the generated communication cycle identifier (CycleID); and

based on the search result, standing by until the searched service description is deleted.

30. (Currently Amended) The method of ~~one of~~ claims ~~28 1, 12, and 17~~, further comprises the steps of:

after the primitive is received, setting a data reception function to a disable state; and
generating an event message including an event code (EventCode) and a state variable (StateVariable) in the primitive, wherein the event message is included in the composing APDU.

31. (Canceled)

32. (Previously Presented) The method of claim 30 further comprises the step of:
after the service description is deleted, setting the data reception function to an enable state.

33. (Canceled)

34. (Currently Amended) A data processing method for application layer based on a protocol composed of at least a lower layer and an application layer, the method comprising the steps of:

receiving a request message receiving primitive (ReqMsgRcv) including an application layer protocol data unit (APDU) from the lower layer;

~~based on a value of an application layer option (ALO) field in the APDU,~~ making a first decision regarding whether to process the APDU according to a value of an application layer option (ALO) field in the APDU;

based on a result of the first decision, extracting a message from the APDU;

making a second decision regarding whether to generate a service description according to the request message receiving primitive (ReqMsgRcv);

based on a result of the second decision, generating the service ~~description~~description;

generating a user request receiving primitive (UserReqRcv) primitive including the message; and

transmitting the generated user request receiving primitive (UserReqRcv) to an application software.

35. (Previously Presented) The method of claim 34 further comprises the step of:
after the primitive receiving step, setting a data reception function to an disable state.

36. (Canceled)

37. (Canceled)

38. (Previously Presented) The method of claim 34, wherein the second decision is made on the basis of a network layer service (NLService) included in the request message receiving primitive (ReqMsgRcv).

39. (Previously Presented) The method of claim 38, wherein if, in the second decision step, the network layer service (NLService) is acknowledged, the service description is generated, and if the network layer service (NLService) is non-acknowledged, the service description is not generated.

40. (Currently Amended) The method of claim 34, wherein the service description comprises at least one ~~of~~among an application layer service code (ALSvcCode), an application layer service type (ALSvcType), a communication cycle number (CycleNo), and a communication cycle identifier (CycleID).

41. - 43. (Canceled)

44. (Currently Amended) The method of claim 35 ~~of claim 39~~ further comprises the step of:

if a network layer service (NLService) of the request message receiving primitive (ReqMsgRcv) is non-acknowledged, setting the data reception function after the transmission of the user request receiving primitive (UserReqRcv) to an enable state.

45. (Canceled)

46. (Previously Presented) A data processing method for application layer based on a protocol composed of at least a lower layer and an application layer, the method comprising the steps of:

receiving a user response sending primitive (UserResSend) from an upper application software;

generating a response message according to the user response sending primitive (UserResSend);

composing an application layer protocol data unit (APDU) including the response message;

transmitting the APDU to the network layer using a pre-stored service description;

and

completing a communication cycle.

47. (Canceled)

48. (Previously Presented) The method of claim 46, wherein the completion step comprises the sub-steps of:

making a first decision regarding whether a service execution time from the network layer is within a service time out (SvcTimeOut); and

based on a result of the first decision, making a second decision regarding whether to receive a network layer completing primitive (NLCompleted).

49. (Previously Presented) The method of claim 48 further comprises the sub-step of: deleting the service description.

50. (Currently Amended) The method of claim 48 ~~or claim 49~~, wherein if, in the first decision step, the service execution time is for the service time out (SvcTimeOut), the second decision step is performed, and if not, the service description is deleted;

wherein if, in the second decision step, the network layer completing primitive (NLCompleted) is received, the service description is deleted, and if not, the first decision step is performed.

51. (Previously Presented) The method of claim 49 further comprises the step of: after the service description is deleted, setting the data reception function to an enable state.

52. -53. (Canceled)

54. (Currently Amended) A data processing method for application layer based on a protocol composed of at least a lower layer and an application layer, the method comprising the steps of:

receiving a message receiving primitive (MsgRev) including an application layer protocol data unit (APDU) from the lower layer;

~~based on a value of an application layer option (ALO) field in the APDU~~, making a first decision regarding whether to process the APDU;

based on a result of the first decision, extracting a message from the APDU;

making a second decision regarding whether a corresponding service description exists;

based on a result of the second decision, generating a predetermined primitive; and transmitting the generated primitive to an upper application software.

55-56. (Canceled)

57. (Previously Presented) The method of claim 54, wherein if, in the second decision step, a service description corresponding to a communication cycle identifier (CycleID) included in the message receiving primitive (MsgRcv) is searched, the message receiving primitive (MsgRcv) comprises a response message.

58. (Canceled)

59. (Previously Presented) The method of claim 54 further comprises the step of:
making a third decision regarding whether the extracted message is an event message,
wherein if, in the second decision step, a service description corresponding to the communication cycle identifier (CycleID) included in the message receiving primitive (MsgRcv) is not searched, the third decision step is performed.

60. - 63. (Canceled)

64. (New) The method of claim 17, further comprises the step of:
after the service description is deleted, transmitting an application layer completing primitive (ALCompleted) to the application software.

65. (New) The method of claim 17, further comprises the steps of:
after the communication cycle identifier (CycleID) is generated, searching a pre-stored service description corresponding to the generated communication cycle identifier (CycleID); and
based on the search result, standing by until the searched service description is deleted.

66. (New) The method of claim 39 further comprises the step of:
if a network layer service (NLService) of the request message receiving primitive (ReqMsgRcv) is non-acknowledged, setting the data reception function after the transmission of the user request receiving primitive (UserReqRcv) to an enable state.

67. (New) The method of claim 49, wherein if, in the first decision step, the service execution time is for the service time out (SvcTimeOut), the second decision step is performed, and if not, the service description is deleted;

wherein if, in the second decision step, the network layer completing primitive (NLCompleted) is received, the service description is deleted, and if not, the first decision step is performed.

68. (New) The method of claim 54, wherein the first decision is made depending on whether a value of an application layer option (ALO) field of the APDU comprises a predetermined value.

69. (New) The method of claim 68, wherein the message is extracted using a value of an APDU header length (AHL) field in the APDU.

70. (New) The method of claim 59, wherein if, in the third decision step, the extracted message is an event message, a user event receiving primitive (UserEventRcv) is generated in the primitive generation step, if the extracted message is not an event message, ignoring the extracted message.

71. (New) The method of according to claim 59, wherein the third decision is made by comparing and deciding whether a command code of the extracted message is equal to a predetermined value of the event message.